A tripartite tensor decomposition fold-in for social tagging

Liao, Zhi-fang; Cai, Fei ; Zhang, Miao; Liao, Zhi-Ning ; Zhang, Yan

Published in:
Journal of Applied Science and Engineering

DOI:
10.6180/jase.2014.17.4.03

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

Journal of Applied Science and Engineering, Vol. 17, No. 4, pp. 363–370 (2014) *DOI: 10.6180/jase.2014.17.4.03*

A Tripartite Tensor Decomposition Fold-in for Social Tagging

Zhi-fang Liao[1]*, Fei Cai[1], Miao Zhang[2], Zhi-ning Liao[3] and Yan Zhang[4]

[1]*School of Software, Central South University, Changsha, Hunan, P.R. China*
[2]*Department of Computer Science, University of Texas at Arlington, Arlington, TX, USA*
[3]*Faculty of Engineering, Science & The Built Environment, London South Bank University, London, UK*
[4]*Institute of Human Development, The University of Manchester, UK*

Abstract

The tripartite tensor decomposition (TTD) model reveals the latent relationship among items, tags and users in social tagging systems in terms of a low order tensor obtained from the high-index sparse data space with the tensor dimensionality reduction technique. The Tripartite decomposition recommendation algorithms can produce high quality recommendations, but have to undergo expensive tensor decomposition steps when new users, new tags, or new items come in, which is significant in light of the tremendous growth in numbers of users, tags and items. In this paper, we present fold-in algorithms for Tripartite tensor decomposition to deal with the new users problem. We evaluate the fold-in algorithms experimentally on several datasets and the results demonstrate the effectiveness of the algorithm.

Key Words: Recommender System, Tripartite Tensor Decomposition, Fold-in, Social Network

1. Introduction

Social tagging is one of the important features of Web 2.0, which enables many users to add metadata in the form of keywords to share items. If a recommendation system enables users to add tags to items, such as adding music features to last.fm, pictures to flickr and web pages to del.icio.us, we call this recommendation system a social tagging system [1].

Social tagging system includes three entities: an item, a tag and a user. An Item is a resource, a user is a client and A tag is a label. Some researchers focus on the problem of presenting the three dimensional relationship in social tagging system.

Though the traditional recommendation algorithms can be used in tagging systems, the accuracy is not good enough as they cannot explain the three dimensional relationship well. But Tensor model is a good way to describe three dimensional structure [2]. Because of the high accuracy of matrix decomposition method for the traditional recommenders, P. Symeonidis et al. [2] first transferred the item-tag-user graph to third-order tensor to represent the social tagging data, and applied a tensor decomposition method (HOSVD) to predict future tagging activities. Afterwards, much research work using tensor decompositions on tag recommendations emerged, such as Reference [3–6] and these methods get good accuracy in recommendation.

However, with the development of social tagging system, a large number of people register as new users, and subsequently new items and new tags also appear very quickly. So the tensor representing the dataset has to be updated frequently. Accordingly, to do tag recommendation towards those new users, new items and new tags, tensor decomposition has to be recalculated, which is a both time-consuming and expensive task. Symeonidis et al. [6] introduced an incremental SVD method to insert new users, and because there are three dimensions, they have to update three-mode SVD to get a new user folded in. Miao Zhang et al. [7] proposed LOTD (low-order ten-

*Corresponding author. E-mail: zfliao@csu.edu.cn

sor decomposition) method to get a new user fold-in.

In this paper, we propose tripartite tensor decomposition method [8] for fold-in, and it will be compared with Tucker decomposition (Tucker) [9], parallel factor decomposition (ParaFac) [10] fold-in methods. All of them can update the factors which are needed to be updated when new users are folded in by matrix operation. We mainly consider the new users fold-in, and new items, new tags can be fold-in in the same way.

The rest of this paper is organized as follows: section 2 summarizes previous work. Section 3 introduces tripartite tensor decomposition model for social tagging mainly from the Ref [8], which helps to describe the fold-in problem clearly. Section 4 presents tripartite fold-in model and algorithm. Section 5 shows our experiment results on three datasets and the comparison between other fold-in tensor decomposition methods. Summary and conclusion are made in section 6.

2. Related Work

In this section, we briefly introduce some research work regarding tag recommendation. To recommend tags to users, Jaschke et al. [11] proposed a simple method to find the popular tags around the user-item pairs and recommend these tags to the users. This method has become the baseline of the tag recommendation method.

By constructing two bipartite item-user graphs, many traditional recommendation systems use collaborative filtering to recommend the items based on similar user preference. Because of the three dimensional structure of the item-tag-user graph, collaborative filtering cannot be directly used for the tag recommendation system. Jaschke et al. [11] reduce the ternary relation to three 2-dimensional projections. They apply collaborative filtering for each bipartite graph and combine these together. Tag-aware Fusion [12] is another tag recommendation method based on the collaborative filtering method. The same as [13], they reduce the three-dimensional relationship to three two-dimensional correlations and associate the relationship based on a fusion method. Because of missing the interactions between three dimensions, the performance of this method is generally lower than tensor methods. Xu et. al. [14] proposes a tag prediction approach based on the HITS (hyperlink-induced topic search) algorithm. However, the similar problem also exists in this algorithm.

But tensor model is good to describe three dimensional structure, and also because of the high accuracy of matrix decomposition method for the traditional recommender, Symeonidis [2,6] first transfers the item-tag-user graph to the tensor structure and then use the tensor decomposition method to predict the tag sets. However, because of the extreme sparsity of real application dataset, it shows that this method does not achieve a high accuracy. Rendle et al. [5] propose a special case of the tucker decomposition model, pairwise interaction model, to predict the tag sets. In their experiments on real world datasets, it is shown that model achieves better prediction quality. Pairwise interaction tensor model is in fact a special case of the 2nd order tensor decomposition [5].

Most of previous work on tensor decomposition applied in social tagging is static. They do not consider the dynamic growth of items, tags and users. When new users come in, the content of tensor expands accordingly. The original tensor decomposition process has to go through again to acquire the predicted values. Obviously, this process is time consuming and costly. Seeing that, we propose fold-in method aiming to solve the new user problem, without re-decomposing the original tensor.

3. TTD Model for Social Tagging

3.1 Tripartite Model

Figure 1 shows tripartite graph where the users are one type of graph nodes with index k, tags are another type of nodes with index j, and items are third type of nodes with index i.

The connection between item node and tag node j is U_{ij}, which is the co-occurrence of item i and tag j. The connection between tag node j and user node k is V_{jk}, which is the co-occurrence of tag i and user k. The connection between item node i and user node k is W_{ik}, which is the co-occurrence of item i and user k.

We define Tripartite model [8] as Eq. (1),

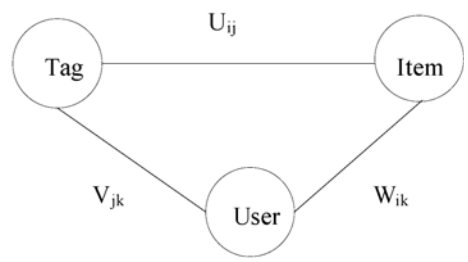

Figure 1. Tripartite graph model.

$$Y_{ijk} = U_{ij} + V_{jk} + W_{ik} \tag{1}$$

Y_{ijk} is a tripartite graph model for social tagging which is appealing because it is intuitive and can be visualized easily. More details for tripartite graph model can refer to [8].

3.2 Tripartite Tensor Decomposition Model

The detail information of nodes' relationship are shown in Figure 2, item nodes are denoted as i_1 through i_4; tag nodes are denoted as j_1 through j_4;user nodes are denoted as k_1 through k_4. Co-occurrence of items and tags are contained in U_{ij}, Co-occurrence of items and users are contained in V_{jk}, and Co-occurrence of items and tags are contained in W_{ik}.

However, there are several crucial drawbacks when using tripartite graph to model the social tagging prediction problem. This is illustrated in Figure 2. Let us consider the co-occurrence between item i_2 and tag j_2. In tripartite graph model, this co-occurrence is defined to be the direct connection of node i_2 and node j_2, which is contained in $U_{i_2 j_2}$. However, this is inadequate. The co-occurrence can also be achieved by the connection from node i_2 to k_2 and k_2 to j_2, which is contained in $W_{i_2 k_2} V_{j_2 k_2}$.

Similarly, consider the co-occurrence between i_3; j_2 in Figure 2. Although there is no connection between i_3; j_2, this co-occurrence is not zero, because of the existence of edges $i_3 - k_2$; $k_2 - j_2$. These two examples show the inadequacy of counting co-occurrence using tripartite graph model.

Considering the drawbacks of tripartite graph mentioned above, we construct a tripartite tensor decomposition model, the details of the model can refer to [8], we propose the following tripartite graph inspired tensor decomposition. We use Y_{ijk} to approximate the tensor X_{ijk},

$$Y_{ijk} \approx X_{ijk} \tag{2}$$

Different Y_{ijk} means different tensor decomposition method. In this paper, we propose the following tripartite graph inspired tensor decomposition:

$$Y_{ijk} = U_{ij} V_{jk} + V_{jk} W_{ik} + U_{ij} W_{ik} \tag{3}$$

To obtain the optimal solution, we get the objective function for Tensor Tripartite Decomposition Framework.

$$\min_{U,V,W} J = \min \sum_{ijk} (X_{ijk} - Y_{ijk})^2 + \alpha(\|U\|^2 + \|V\|^2 + \|W\|^2) \tag{4}$$

J the optimal value of tensor X_{ijk}, α is a model parameter to regularize U, V, W such that elements of U, V, W have nearly same magnitude.

3.3 Fold-in Problem

For most social tagging systems, such as del.icio.us, Facebook etc., thousands of people register as new users every day. When these new users log on the system, they perhaps provide some preferences upon items and tags. To recommend tags to those new users, we need to consider both the history of tagging activities of existing users and the preference information of new users.

Although the most commonly encountered situation is the addition of newly joined users, other forms of updating also occur; for example, new tags could be introduced as users demand; or the number of items could be increased as variety and/or social media contents become richer, more diverse, etc. In this paper we study the case when the numbers of users increase. But the same method can be easily extended to the increase of tags or items.

We represent existing data as

$$X^{old} = (X_1, X_2, \cdots, X_{n_k})$$

and new data with partial user preferences as

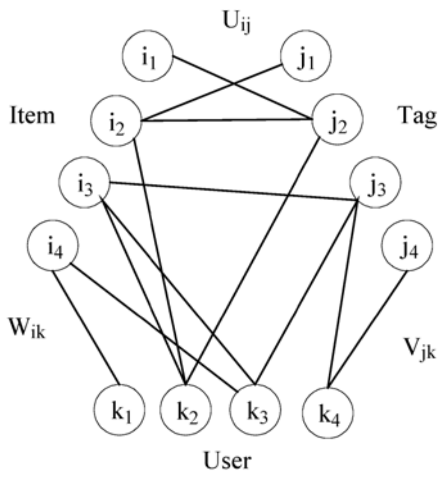

Figure 2. Tripartite graph for social tagging. Co-occurrence of items and tags are contained in U_{ij}. Similar for V_{jk}; W_{ik}. Note that co-occurrence between i_2; j_2 are not only achieved by the direct edge $i_2 - j_2$ contained in U, but is also achieved through the edges $i_2 - k_2$, $k_2 - j_2$ contained in VW_T. Co-occurrence between i_3; j_2 is not zero, because of the existence of edges $i_3 - k_2$; $k_2 - j_2$.

$$X^{new} = (X_{n_k+1}, X_{n_k+2}, \cdots, X_{n_k+l})$$

the complete information for old users and new users can be represented as

$$X = (X^{old}, X^{new}) = (X_1, X_2, \cdots, X_{n_k}, X_{n_k+1}, X_{n_k+2}, \cdots, X_{n_k+l})$$

X^{old} represents the history of social tagging activities, X^{new} represents the new users and their preference information, some of the tensor factors need updating. As we can see in Figure 3, the number of tensor factors needed to be updated are fortunately small, in fact only one factor k, need to be updated when new users register in tagging system. Therefore, fold-in techniques can be adopted to accomplish tag recommendation for new users. In the following, we outline the tripartite tensor decomposition fold-in model in section 4.

4. TTD Fold-in Algorithm

As mentioned in section 3.3, a large number of new users log on social tagging systems every day. To deal with the problem of recommending personalized tags to those new users efficiently, we propose fold-in method for tripartite tensor decomposition. This paper focuses on fold-in new users into the system.

Tensor factors of X^{old} are computed using model in [8]. Based on those factors, we can fold in X^{new} without decomposing $X = (X^{old}, X^{new})$ all over again to get the prediction values of X^{new}.

For TTD the new user fold-in shaded in Figure 3, the shaded part of X represents new users X^{new}. Among model parameters, U remains unchanged, and the size of V will change from $n_j \times n_k$ to $n_j \times (n_k + l)$, the size of W will change from $n_i \times n_k$ to $n_i \times (n_k + l)$ where l is the number of new users. Then we split V and W as following.

V changes as $V_{n_j \times (n_k+l)} = (V^{old}_{n_j \times n_k}, V^{new}_{n_j \times l})$, then

$$\begin{aligned} V^{old} &= (V_1, V_2, \cdots, V_{n_k}) \\ V^{new} &= (V_{n_k+1}, V_{n_k+2}, \cdots, V_{n_k+l}) \end{aligned} \tag{5}$$

W changes as $W_{n_i \times (n_k+l)} = (W^{old}_{n_i \times n_k}, W^{new}_{n_i \times l})$, then

$$\begin{aligned} W^{old} &= (W_1, W_2, \cdots, W_{n_k}) \\ W^{new} &= (W_{n_k+1}, W_{n_k+2}, \cdots, W_{n_k+l}) \end{aligned} \tag{6}$$

As V^{old} and W^{old} do not changed, so what we need to do is to compute V^{new} and W^{new}.

4.1 Algorithm

Substituting $X = (X^{old}, X^{new})$ and $V = (V^{old}, V^{new})$, $W = (W^{old}, W^{new})$, while fixing the old parameters, the optimal X^{new}, V^{new}, W^{new} is obtained by the following algorithm.

(1) Get the new user objective function

$$X_{ijk}^{new} = U_{ij} V_{jk}^{new} + V_{jk}^{new} W_{ik}^{new} + U_{ij} W_{ik}^{new} \tag{7}$$

(2) V^{new}, W^{new} can be obtained in Eq. (8).

$$\begin{aligned} \min_{ijk} J_{new} = \min \sum_{ijk} [X^{new} - (U_{ij} V^{new} + V^{new} W^{new} + U_{ij} W^{new})]^2 \\ + \alpha \sum_{jk} (V^{new})^2 + \alpha \sum_{ik} (W^{new})^2 \end{aligned} \tag{8}$$

When processing the first new user, setting $k = n_k + 1$, V^{new} and W^{new} are vectors and described as v, w.

$$v = \begin{bmatrix} v_{1,n_k+1} \\ v_{2,n_k+1} \\ \vdots \\ v_{n_j,n_k+1} \end{bmatrix} \quad w = \begin{bmatrix} w_{1,n_k+1} \\ w_{2,n_k+1} \\ \vdots \\ w_{n_i,n_k+1} \end{bmatrix}$$

Dv is the diag of v, and Dw is the diag of w.

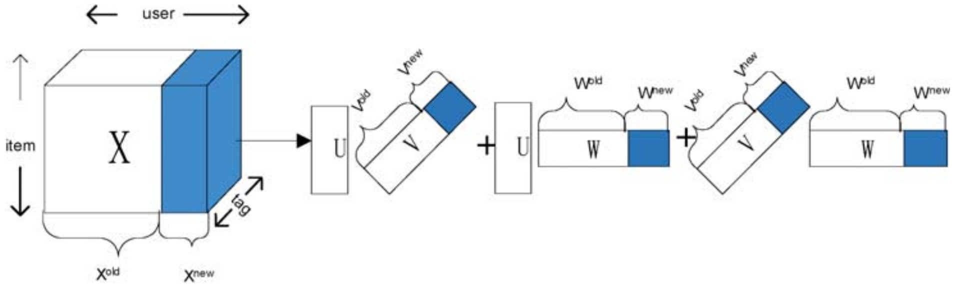

Figure 3. TTD new user fold-in.

$$D_v = \begin{bmatrix} v_{1,n_k+1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & v_{n_j,n_k+1} \end{bmatrix}$$

$$D_w = \begin{bmatrix} w_{1,n_k+1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{n_i,n_k+1} \end{bmatrix}$$

To get the new user, we need to get the derivative of D_v, D_w, v, w following the Eq. (9).

$$J_{new}(D_v, D_w) = [X_{ij}^{new} - (U_{ij}D_v + D_w U_{ij} + V_{jk}W_{ik})]^2 + \alpha V_{jk}{}^2 + \alpha W_{ik}{}^2 \tag{9}$$

To obtain the optimal of D_v, D_w, v, w, set $\partial J / \partial D_v = 0$, $\partial J / \partial D_w = 0$, $\partial J / \partial v = 0$, $\partial J / \partial w = 0$, get the optimal values of V^{new} and W^{new} as following.

$$v = [D + (\|W\|^2 + \alpha)]^{-1} C \tag{10}$$

where $D = diag(diag(B) + 2a)$, $C = X^T W - U^T D_w W + diag(U^T X - U^T D_w U)$, $a = U^T W$, $B = U^T U$.
and

$$w = [diag(v)U^T + UD_v + D + (\| v \|^2 + \alpha)I]^{-1}[X_v - UD_v v + diag(XU^T - UD_v U^T)] \tag{11}$$

where $D = diag(d)$, $d = diag(B)$, $B = UU^T$.

Algorithm: TTD New User Fold-in

1: Input:
  X: The initial tensor data;
  U: The connection between item and tag, computed by TTD.
2: Output:
  V^{new}: The optimal value of ;
  W^{new}: The optimal value of W;
  X_p^{new}: The predicated tensor of X^{new};
3: Initialize X^{new}, the tensor of the new user;
4: Initialize V^{new} and W^{new} based on X^{new};
5: Do $t = 0$ to max-iteration
6: update V^{new} by Eq. (10);
7: update W^{new} by Eq. (11);
8: input X^{new}, V^{new} and W^{new} to Compute Eq. (8) as J_{new}^t;
9: if $\|J_{new}^{t+1} - J_{new}^t\| < \varepsilon$, go to Line 12;
10: End Do;
11: Compute $X_p^{new} = U_{ij}V^{new} + V^{new}W^{new} + U_{ij}W^{new}$;
12: return X_p^{new};
13: End

The detail of TTD fold-in algorithm is shown as follows.

5. Experiments Results

We carry out experiments on several real world datasets to show and evaluate the performance of TTD fold-in method. Section 5.1 presents the datasets and experimental setting. Section 5.2 the evaluation strategy and section 5.3 reports the performance results by Top N precision-recall curves and the parameters in the experiments.

5.1 Datasets

In Table 1, 2, we list the statistics of three social tagging recommendation datasets. The detail information on experiment data sets can be found in Table 1. Each dataset can be presented as a tensor with the size of item × tag × user.

5.1.1 Last.fm

Dataset consists of web pages crawled from web site which is a social tagging systems providing the personalized media for users, the system also permit users add tags on the system. We choose the active users which mark more than 2400 tags but less than 5000 times on the items, then the tags with 1000 times but less than 4000 times and the items which is marked more than 88 times but less than 3000 times in the dataset crawled in the first half of 2009.

5.1.2 Bibsonomy

Dataset downloads from bibsono-my.org. We first choose the active users which mark more than 1000 tags but less than 2600 times on the items, the tags which is used by 420 times but less than 2000 times and the items

Table 1. Dataset A

Dataset	Item	Tag	User
Last.fm	1000	1570	28000
Movielens	980	2000	24600
Bibsonomy	1600	1010	20400

Table 2. Dataset B

Dataset	Item	Tag	User
Last.fm	2030	2410	42500
Movielens	3450	3690	46500
Bibsonomy	3620	1160	36100

which is marked more than 76 times but less than 1000 times.

5.1.3 MovieLens

Dataset collects the item-tag-user information from the online movie recommender system. We choose the active users which mark more than 30 tags but less than 600 times on the items, tags used by 30 times but less than 1000 times and the items which is marked more than 25 times but less than 1000 times.

To demonstrate the influence of different densities of datasets on the performance of TTD fold-in method, we choose two subsets – A and B with different densities from each real world datasets. The size of subset B is relatively bigger than subset A of each dataset. And B is sparser than the corresponding subset A. For each dataset, we list the statistics of each subset in the following. As an example, subset A of last.fm is about 1000 items, 1570 tags and 28000 users, subset B of last.fm is about 2030 items, 2410 tags and 42500 users, the NNZ of subset A is 0.23, and subset B is 0.15.

5.2 Evaluation Strategy

We use the similar evaluation protocol in [3]. 10-fold cross-validation is adopted in all experiments. For each dataset listed above, we randomly partition the input tensor into 10 parts. Each part is retained as the testing data, which can be called as fold-in tensor X_{real}^{new}. All the other parts of input tensor constitute a training tensor, which is called the tensor X^{old}. For each time, we use X^{old} to predict the tensor X_{real}^{new} and the average of precision and recall results is the final prediction result.

For each fold-in prediction, we use traditional Precision-Recall methods in a top-N fashion. For each post(i; k), we sort the predicted values. We pick $N = 1, 2, 3, \ldots, 10$ top values and return the corresponding tags associated with these picked values. We assess the returned tags with the known information which have been masked out. The precision and recall are defined as [3].

5.3 Performance of the Proposed Algorithms

We compare the prediction qualities of three fold-in algorithms as Tucker [9], Parafac [10] and TTD fold-in.

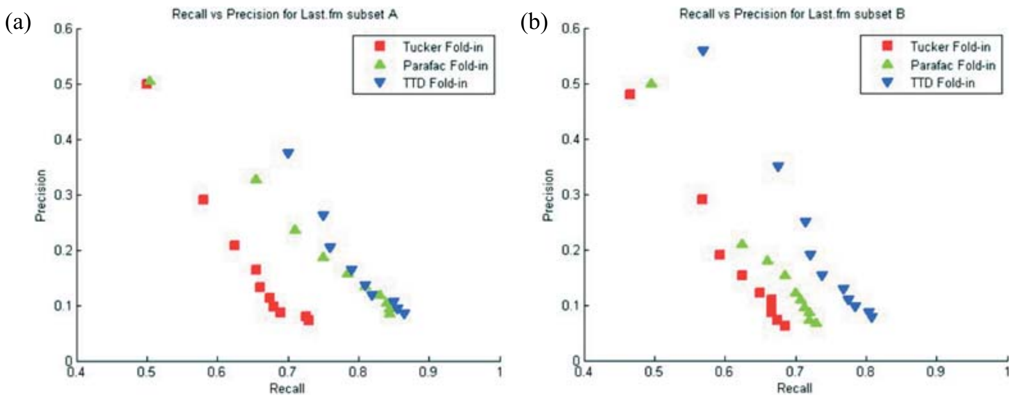

Figure 4. (a) Last.fm data: subset A. (b) Last.fm data: subset B.

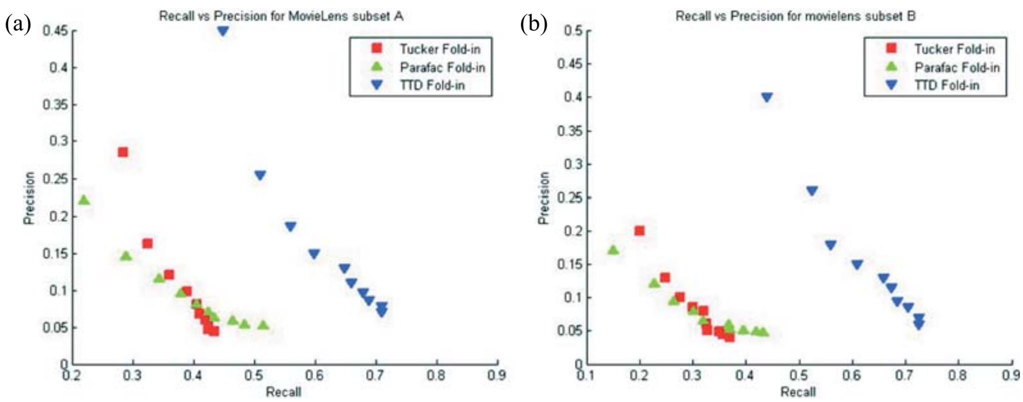

Figure 5. (a) Movielens: subset A. (b) Movielens: subset B.

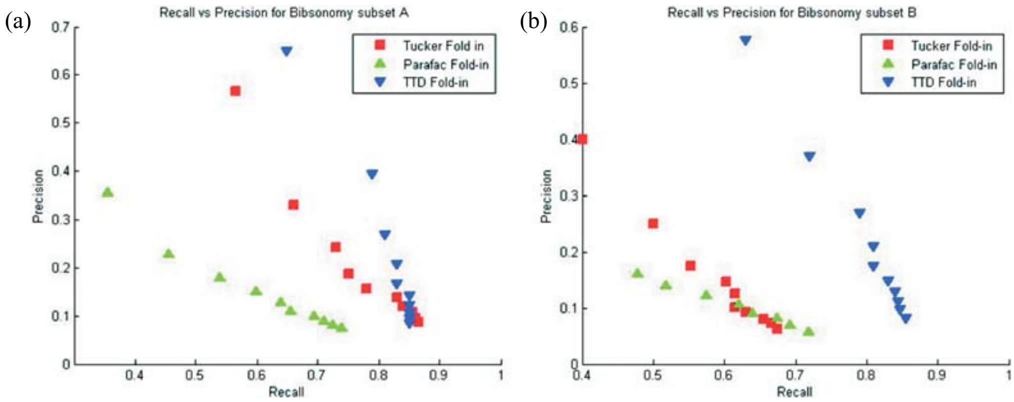

Figure 6. (a) Bibsonomy: subset A. (b) Bibsonomy: subset B.

Figure 4(a)–6(a) shows the comparison between (1) Tucker Fold-in, (2) ParaFac Fold-in, (3) TTD Fold-in on subset A of each dataset, and Figure 4(b)–6(b) shows the comparison between the three Fold-in methods on subset B.

Figure 7 shows the comparison between (1) Tucker Fold-in, (2) ParaFac Fold-in, (3) LOTD (2D) fold-in and (4) TTD Fold-in on subset A of Bibsonomy dataset.

The experiment results indicate: (A) TTD Fold-in method has better precision-recall curves for all datasets than Tucker and Parafac fold-in because of the sparsity of real world dataset, and nearly the same precision except for the Top 1 and Top 2 new user predication when compares with LOTD (2D) with Bibsonomy as the two models have the same ability to deal with sparse data. (B) Tucker and ParaFac fold-in methods are based on the traditional tensor decomposition models, but the accuracy of these two methods is lower than TTD fold-in model as they over-fit these sparse tensor data. (C) For each dataset, we notice that the difference between TTD fold-in and Tucker/ParaFac fold-in methods on subset A is much bigger than that of subset B, which means that TTD fold-in can gain relatively much better performance than the other.

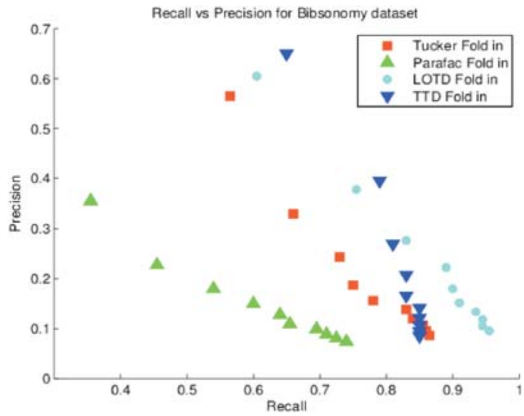

Figure 7. Four methods comparison on bibsonomy: subset A.

6. Conclusions

When the number of users is large enough, and the number of new users is relatively small, that is to say, $l < n_k$, the fold-in methods can provide prediction quality, which is consistent with our intuition. The tensor factors coming from X^{old} carry the historic information and also the similar users activities. While factors from X^{new} indicate the new users' personalized preference and features those can help to detect their potential activities. When l *gets bigger, social tagging system can combine current* X^{new} into X^{old}, and then reproduce the decomposed factors. With new decomposed factors, which carry more latest information and trends, the system can do online recommendations again.

The fold-in techniques proposed in this paper have fast online performance, requiring just a few simple matrix operations for new users. Meanwhile, the experiment results demonstrate that the fold-in methods can provide comparable prediction quality.

Especially, TTD fold-in method based on tripartite tensor decomposition model specifically targets at the sparsity challenge in tag recommendation systems, because low-order polynomials can enhance the statistics of the sparse tagging datasets. Therefore, it can gain better predicting accuracy than the other two fold-in methods, which are based on two traditional tensor decomposition models. The traditional tensor methods (Tucker and ParaFac) obviously overfit the sparse tensor decompositions. The fold-in methods can help social tagging recommendation systems achieve high scalability while providing good predictive accuracy.

Acknowledgements

This work is supported by the Hunan NSF (12JJ 3073), Hunan Science Project (2012GK3170) and National Science Project (2012BAH08B01).

References

[1] Adomavicius, G. and Tuzhilin, A., "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, pp. 734–749 (2005). doi: 10.1109/TKDE.2005.99

[2] Symeonidis, P., Nanopoulos, A. and Manolopoulos, Y., "Tag Recommendations Based on Tensor Dimensionality Reduction," Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 43−50, ACM (2008). doi: 10.1145/1454008.1454017

[3] Cai, Y., Zhang, M., Luo, D., Ding, C. and Chakravarthy, S., "Low-Order Tensor Decompositions for Social Tagging Recommendation," Proceedings of the 2011 ACM International Conference on Web Search and Data Mining, pp. 695−704 (2011). doi: 10.1145/1935826.1935920

[4] Rendle, S., Balby Marinho, L., Nanopoulos, A. and Schmidt-Thieme, L., "Learning Optimal Ranking with Tensor Factorization for Tag Recommendation," Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 727–736, ACM (2009). doi: 10.1145/1557019.1557100

[5] Rendle, S. and Schmidt-Thieme, L., "Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation," Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 81−90, ACM (2010). doi: 10.1145/1718487.1718498

[6] Symeonidis, P., Nanopoulos, A. and Manolopoulos, Y., "A Unified Framework for Providing Recommendations in Social Tagging Systems Based on Ternary Semantic Analysis," *IEEE Transactions on Knowledge and Data Engineering* (2009). doi: 10.1109/TKDE.2009.85

[7] Zhang, M., Ding, C. and Liao, Z. F., "Tensor Fold-in Algorithms for Social Tagging Prediction," 2011 11th IEEE International Conference on Data Mining, pp. 1254−1259 (2011). doi: 10.1109/ICDM.2011.142

[8] Liao, Z. F., Li, L., et al., "A Tripartite Decomposition of Tensor for Social Tagging," *Chinese Journal of Computers*, Vol. 35, No. 12, pp. 2625−2632 (2012).

[9] Lathauwer, L. D., Moor, B. D. and Vandewalle, J., "A Multilinear Singular Value Decomposition," *SIAM Journal on Matrix Analysis and Applications*, Vol. 21, pp. 1253−1278 (2000). doi: 10.1137/S0895479896305696

[10] Tomasi, G. and Bro, R., "PARAFAC and Missing Values," *Chemometrics and Intelligent Laboratory Systems*, Vol. 75, No. 2, pp. 163−180 (2005). doi: 10.1016/j.chemolab.2004.07.003

[11] Jaschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L. and Stumme, G., "Tag Recommendations in Folksonomies," *Knowledge Discovery in Databases: PKDD 2007*, pp. 506–514 (2007). doi: 10.1007/978-3-540-74976-9_52

[12] Tso-Sutter, K., Marinho, L. and Schmidt-Thieme, L., "Tagaware Recommender Systems by Fusion of Collaborative Filtering Algorithms," Proceedings of the 2008 ACM Symposium on Applied Computing, pp. 1995−1999, ACM (2008). doi: 10.1145/1363686.1364171

[13] Hotho, A., Jaschke, R., Schmitz, C. and Stumme, G., "Information Retrieval in Folksonomies: Search and Ranking," *The Semantic Web: Research and Applications*, pp. 411–426 (2006). doi: 10.1007/11762256_31

[14] Xu, Z., Fu, Y., Mao, J. and Su, D., "Towards the Semantic Web: Collaborative Tag Suggestions," Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland (2006).

Manuscript Received: Mar. 6, 2013
Accepted: Sep. 22, 2014